

变邻域分解多目标自适应差分进化算法

刘志君^{1†}, 高亚奎², 章卫国¹, 王晓光¹, 袁燎原¹

(1. 西北工业大学 自动化学院, 陕西 西安 710129; 2. 中国航空工业第一飞机研究院, 陕西 西安 710000)

摘要: 分解方法是处理复杂问题常用的一种手段, 而差分进化算法被广泛地应用于多目标优化问题 (multi-objective optimization problems, MOP), 为了克服经典差分进化算法和分解方法的缺陷, 本文提出了一种自适应差分进化算法和变邻域分解方法相结合的新颖算法—ADEMO/D-ENS, 该算法采用Tchebycheff方法将多目标优化问题分解成多维标量优化子问题, 并利用邻域子问题的信息进行优化, 基于邻域种群集依概率自适应选择邻域种群规模; 同时采用概率匹配(probability match, PM)自适应方法从差分策略池中选择差分进化策略; 同时分析了算法的复杂度; 最后, 通过和经典的非支配排序遗传算法(non-dominated sorting genetic algorithms II, NSGA-II)和多目标差分进化算法(multi-objective differential evolution algorithm, MODE)仿真对比, 说明ADEMO/D-ENS方法可以更有效的处理多目标优化问题.

关键词: 分解; 邻域种群集; 概率匹配方法; 差分进化; 多目标优化; 复杂度分析

中图分类号: TP301.6 文献标识码: A

Decomposition with ensemble neighborhood size multi-objective adaptive differential evolutionary algorithm

LIU Zhi-jun^{1†}, GAO Ya-kui², ZHANG Wei-guo¹, WANG Xiao-guang¹, YUAN Liao-yuan¹

(1. Automation School, Northwestern Polytechnical University, Xi'an Shaanxi 710129, China;

2. The First Aircraft Institute of China Aviation Industry Corporation, Xi'an Shaanxi 710000, China)

Abstract: Decomposition is a conventional optimization method, and the differential evolutionary algorithm is widely applied in the multi-objective optimization problems (MOP). A novel algorithm—ADEMO/D-ENS which combines the two algorithms, the adaptive differential evolutionary algorithm and the decomposition with variable neighborhood size, is proposed to overcome the drawbacks of the classical differential evolution algorithm and the decomposition method. The approach makes use of the Tchebycheff method to decompose the multi-objective optimization problems into scalar optimization sub-problems. And the sub-problems are optimized by neighborhood relations among them. The adaptive selection approach based on ensemble of neighborhood size is used to determine the neighborhood size. Meanwhile, the probability match adaptive method is used to select differential strategy from the differential strategy pool. Moreover, the complexity of the algorithm is analyzed. Finally, compared with the classical non-dominated sorting genetic algorithms II (NSGA-II) algorithm and the multi-objective differential evolution algorithm (MODE), simulation results verified that the ADEMO/D-ENS approach can deal with the multi-objective optimization problems more effectively.

Key words: decomposition; ensemble neighborhood size; probability matching method; differential evolution; multi-objective optimization; complexity analysis

1 引言(Introduction)

多目标优化问题(multi-objective optimization problems, MOP)一般有几个目标, 而通常彼此之间相互冲突, 与单目标优化不同的是, 多数情况下, 多目标优化中某个目标的改善可能引起其他目标性能的下降, 因此, 多目标优化不可能使所有目标同时达到最优, 只能在各目标间进行协调权衡并折中处理, 使所有目标尽可能达到最优.

分解法是处理复杂问题常用的一种手段, 但在多目标优化中没有得到广泛应用, 主要有两种可能^[1]: 1) 该方法仅定义一个具体的目标进行优化, 其他目标作为约束处理; 2) 将不同的目标通过一个权重和的形式用一个聚合一维函数表达. 近年来, 随着群体智能优化算法的发展, 越来越多的被用于多目标优化. 这些方法得到的有效解称为Pareto最优解集(PS), 该解集和目标空间的Pareto前沿(pareto front, PF)直接相

关^[2]. 但在许多多目标优化的实际应用中, PF的近似需要决策者选择一个最终的优先解, 而大多数MOP可能有许多或无限个Pareto最优向量. 对决策者来说, 得到一个完整的PF是一件很费时的事情.

Zhang等^[3]提出一种基于分解的多目标进化算法, 将多目标优化问题分解成一组标量优化子问题, 利用子问题间的信息对其同时进行优化, 降低了多目标优化的复杂性. 为了提高算法的性能, 不同的多目标问题需要不同的邻域规模(neighborhood size, NS), 甚至某一特定问题的不同搜索阶段需要不同的邻域规模. Zhao等^[4]给出了一个NS集合, 对每一个子问题基于概率动态选择当前子问题的NS.

差分进化作为进化算法的一个分支, 具有有效而强大的搜索能力, 被广泛应用于科学和工程领域^[5-8]. Storn和Price^[9]在2001年将差分进化策略由原来的5种扩展至10种, 近年来, Lampinen和Fan^[10]提出了一种三角变异操作算子改进差分算法的性能, 又有学者给出了一种基于参数向量邻域拓扑结构的新差分变体算法, 该变体算法能够克服经典差分算法的缺陷^[11]. Das等^[12]将差分操作算子引入粒子群算法调整粒子的飞行速度. 以上研究对差分算法的变异操作进行了改进, 使算法的性能得到了改善, 但选择何种差分策略和合适的控制参数需要经过多次试验调整, 这又是一件很耗时的事情.

根据文献[13-16]的自适应调整策略, 本文提出一种新颖的算法—ADEMO/D-ENS, 该方法将基于邻域种群集的分解法和基于PM自适应的差分进化算法相融合用于多目标的优化问题. 自适应差分选择策略有效弥补了经典差分算法需耗费大量时间选择恰当的变异算子和控制参数的问题, 而基于邻域种群集的算法满足不同进化阶段需要不同邻域种群的要求.

2 多目标分解(Multi-objective decomposition)

2.1 多目标优化问题(Multi-objective optimization problem)

多目标优化问题的一般数学模型^[2]如下:

$$\begin{cases} \min/\max F(X) = (f_1(X), \dots, f_m(X))^T, \\ g_i(X) \geq 0, i = 1, 2, \dots, G, \\ \text{s.t. } h_j(X) = 0, j = 1, 2, \dots, H, \\ x_d^L \leq x_d \leq x_d^U, d = 1, 2, \dots, D, \end{cases} \quad (1)$$

其中: $f_1(x), \dots, f_m(x)$ 为目标向量, 可行解向量 $X = (x_1, x_2, \dots, x_D)$, $X \in X$, X 为满足约束的Pareto集合(PS), $g_i(x)$ 和 $h_j(x)$ 分别为不等式约束和等式约束, G 和 H 分别为不等式约束和等式约束个数, x_d^L 和 x_d^U 分别为第 d 决策变量的下界和上界, D 为决策变量维数. 本文仅考虑无约束下的多目标最小化问题.

2.2 分解方法(Decomposition method)

将PF分解成许多标量单目标优化子问题的方法有很多, 常用的有权重和方法、Tchebycheff方法和边界相交方法^[3]. 虽然权重和方法能很好的解决具有凸面或凹面Pareto前沿的优化问题, 但却不能得到非凸(或非凹)Pareto前沿的优化解; 对于目标函数多于两个的优化问题, 边界相交方法得到的最优解比Tchebycheff方法得到的最优解分布更均匀, 但边界相交方法需要设置惩罚因子, 而优化结果的性能对惩罚因子的设定比较敏感. 基于上述因素, 在本文中选择Tchebycheff方法对式(1)分解. 采用Tchebycheff方法对问题(1)进行分解操作, 得到标量优化子问题的形式如下:

$$\begin{cases} \min g_Tch(x/\lambda, z^*) = \max_{1 \leq i \leq m} \{\lambda_i |f_i(x) - z_i^*|\}, \\ \text{s.t. } x \in \Omega, \end{cases} \quad (2)$$

其中: $z_i^* = \min \{f_i(x) | x \in \Omega\}$, $i = 1, \dots, m$, $\lambda = (\lambda_1, \dots, \lambda_m)^T$, 对所有 $i = 1, \dots, m$, $\lambda_i \geq 0$, 且 $\sum_{i=1}^m \lambda_i = 1$, 对每一个Pareto解 x^* 存在一个权向量 λ , x^* 是问题(2)的最优解, 问题(2)的每一个最优解即是问题(1)的Pareto最优解, 因此对多目标优化问题(1)转换成对标量子问题(2)的优化.

3 变邻域自适应差分进化算法 (Adaptive differential evolution algorithm with ensemble population size)

基于分解方法将多目标优化问题(1)分解成 N 维标量优化子问题, 每一个子问题利用其邻域子问题的信息进行优化, 并在当前邻域种群中寻找其最优解, 因此邻域种群大小在算法进化中发挥着关键作用. 不同的优化问题可能需要不同的邻域种群, 甚至同一个优化问题的不同优化阶段也需要不同的邻域种群, 比如当某些解陷入局部最优区域时, 需要较大的邻域种群使其跳出局部最优. 集成学习能够有效的在线自适应调整算法控制参数和算子^[17-18], 本文在自适应差分算法中采用一个NS集合, 基于之前的性能依概率动态选择邻域种群大小.

3.1 差分策略(Differential strategies)

参照文献[1], 本文选用4种差分策略, 如下所示, 其中“DE/nonlinear”策略采用文献[13]中的变异操作算子. 1) “DE/rand/1”; 2) “DE/rand/2”; 3) “DE/nonlinear”:

$$\rho(\gamma) = \gamma^2 \eta_a + \gamma \eta_b + \eta_c, \quad (3)$$

其中 γ 基于插值概率 P_{inter} 产生,

$$\gamma \in \begin{cases} U[0, 2], & \text{如果 } rand \leq P_{inter}, \\ U[2, 3], & \text{其他,} \end{cases}$$

rand为0 - 1之间随机均匀分布的随机数.

$$\begin{cases} \eta_a = \frac{x_c - 2x_b + x_a}{2}, \\ \eta_b = \frac{4x_b - 2x_c - x_a}{2}, \\ \eta_c = x_c, \end{cases} \quad (4)$$

其中 x_a, x_b, x_c 为从当前种群中随机选择的个体;

4) “DE/current-to-rand/1”.

3.2 变邻域自适应差分进化算法 (Adaptive differential evolution algorithm with ensemble population size)

设定多目标优化问题被分解成种群规模为 N 的标量优化子问题, 对每一个子问题 i 计算与其距离最近的 T 个邻域子问题, 并与 T 个邻域子问题组成网络结构, 且集合 T 中的元素相互独立, 子问题 i 利用其邻域子问题的信息进行优化, 而邻域种群规模 T 从邻域种群集中依概率选择. 给定 $C > 1$ 邻域种群数, 并按照邻域种群规模从小到大的顺序组成邻域种群候选池 $NSs = \{N_1, N_2, \dots, N_n\}$, n 为邻域种群集的元素个数, 且 $N_1 \neq N_2 \neq \dots \neq N_n$. 依概率选择每一个优化子问题的邻域种群, 初始各邻域种群具有相同的选择概率, 设置 $p_c = 1/n$. 小的邻域种群满足算法快速收敛的需要, 大的邻域种群帮助算法逃逸局部最优. 邻域种群选择概率更新方式如下:

$$p_{c,G} = \frac{S_{c,g}}{\sum_{c=1}^C S_{c,g}}, \quad (5)$$

其中:

$$S_{c,g} = \frac{\sum_{g=G-LP}^{G-1} W_{-S_{c,g}}}{\sum_{g=G-LP}^{G-1} W_{c,g}} + \varepsilon, \quad c = 1, \dots, C; \quad G > LP, \quad (6)$$

LP 为设定的学习周期, $S_{c,g}$ 表示在前 LP 代内第 c 邻域种群产生的改善解所占的比例, 意味着新解成功进入下一代. $W_{c,g}$ 代表在前 LP 代内第 c 邻域种群产生的所有解的总数, $W_{-S_{c,g}}$ 代表前 LP 代内第 c 邻域种群产生的改善解的总数 (即成功进入下一代的新解个数), 设定一个常值 ε 避免可能出现的零概率.

采用第3.1节给出的4种差分策略组成差分策略池 $S = \{a_1, \dots, a_K\}$ 以及与其相应的策略选择概率向量 $P(g) = \{p_1(g), \dots, p_K(g)\}$, 且满足 $\forall g: p_{\min} \leq p_i(g) \leq 1, \sum_{i=1}^K p_i(g) = 1$, 并设定初始选择概率 $p_k = 1/K$, 其中 K 为差分策略数, g 为当前迭代次数. 采用 PM 方法自适应更新策略 a 的概率 $p_a(g)$, 策略选择概率更新如下^[11-12, 14]:

$$p_k(g+1) = p_{\min} + (1 - K * p_{\min}) * \frac{q_k(g+1)}{\sum_{k=1}^K q_k(g+1)}, \quad (7)$$

其中 $q_k(g+1)$ 表示策略 a 在第 $g+1$ 代的经验估计:

$$q_a(g+1) = q_a(g) + \alpha [r_a(g) - q_a(g)], \quad (8)$$

其中: $r_a(g)$ 表示在第 g 代, 应用策略 a 获得的回报, α 为自适应率. 显然, $\sum_{k=1}^K p_k(g+1) = 1$, 当在一个较长的周期内只采用某一特定策略时, 仅得到该策略的回报值, 其他策略回报值为0, 该策略的选择概率趋近于 $p_{\max} = p_{\min} + (1 - K * p_{\min})$ 为避免有策略的选择概率等于0, 在此设定 $p_{\min} = 0.05$.

可信度赋值 (即策略回报值) 的方法有4种^[11, 15], 根据文献[1]的分析, 本文采用 ExtAbs 方法计算可信度赋值, 表达式如下:

$$r_k(g) = \varsigma_k^*(g) = \max [\varsigma_k^s(g)]_{s=1, \dots, |R_k|}, \quad (9)$$

其中 $\varsigma_k^s(g)$ 为在第 g 代利用第 k 差分策略得到的第 s 相对适应度的改善值, 即

$$\varsigma_k^s(g) = \begin{cases} \frac{f_{\text{best}}}{f_c^s(g)} * |f_p^s(g) - f_c^s(g)|, \\ \text{如果 } f_c^s(g) \text{ 优于 } f_p^s(g), \\ 0, \text{ 其他} \end{cases} \quad (10)$$

其中: f_{best} 为当前种群最优解的适应度值, $f_c^s(g)$, $f_p^s(g)$ 分别为第 g 代子代和父代的适应度值. $R_k(g) = \{\varsigma_k^1(g), \dots, \varsigma_k^{|R_k|}(g)\}$ 为采用第 k 策略在第 g 代获得的改善值的集合, $|R_k|$ 为 R_k 的势.

在算法进化过程中, 对分解后的每一个待优化子问题 i , 基于概率 $p_c = 1/n$ 选择当前子问题的一个邻域种群 T , 并计算距离每一个权向量最近的 T 维向量, 对 $i = 1, \dots, N$, 设置 $A(i) = \{i_1, \dots, i_T\}$, 其中 $\lambda^{i_1}, \dots, \lambda^{i_T}$ 是距离 λ^i 最近的 T 维权向量. 根据设定概率 δ , 确定差分操作范围 P_{size} .

$$P_{\text{size}} = \begin{cases} A(i), & \text{如果 } \text{rand} < \delta, \\ \{1, \dots, N\}, & \text{其他.} \end{cases} \quad (11)$$

对种群 P_{size} , 依概率选择当前子问题的差分操作策略, 对产生的新解 y 用式(12)进行修正得到解 y_m ,

$$y_{m,d} = \begin{cases} y_d + \xi_d \cdot (y_d^U - y_d^L), & \text{概率为 } p_m, \\ y_d, & \text{概率为 } 1 - p_m, \end{cases} \quad (12)$$

其中:

$$\xi_d = \begin{cases} (2 \cdot \text{rand})^{\frac{1}{\tau+1}} - 1, & \text{如果 } \text{rand} < 0.5, \\ 1 - (2 - 2 \cdot \text{rand})^{\frac{1}{\tau+1}}, & \text{其他,} \end{cases}$$

τ 和 p_m 为 DE 参数, $\text{rand} \in U[0, 1]$, y_d^U , y_d^L 分别为第 d 维决策变量的上下界. 若 y_m 有一个目标值优于已存储的 Pareto 最优解, 将 y_m 用于式(2)和 z^* 的更新, 则 y_m 成

功进入下一代; 为了保持种群多样性, 避免子代解在种群 P_{size} 中被多次复制, 设定种群更新最大值 $n_r \ll P_{\text{size}}$, 更新种群 N ; 式(6)更新邻域种群选择概率, 式(10)计算相对适应度值的改善, 得到可信度赋值 $r_k(g)$, 式(7)–(8)更新差分策略选择概率. 达到最大迭代次数, 算法输出 Pareto 近似前沿和 Pareto 占优解.

3.3 复杂度分析(Complexity analysis)

非支配排序遗传算法(non-dominated sorting genetic algorithm II, NSGA-II)是一种基于非支配排序方法的快速精英多目标优化算法, 作为多目标优化算法的优秀代表, 已被成功用于不同多目标问题的优化, 常被用于与其他多目标优化的比较^[2,19].

1) 空间复杂度.

假定 ADEMO/D-ENS 算法和 NSGA-II 算法具有相同的初始种群规模 N , 在每一次迭代过程中, ADEMO/D-ENS 算法需维持 N 个解的内部种群, 需要存储空间为 $O(N)$, NSGA-II 算法在非支配排序过程需要存储空间为 $O(N^2)$, 在拥挤距离计算过程需要存储空间为 $O(N)$, 因此, NSGA-II 算法需要存储空间为 $O(N^2)$. 显然, ADEMO/D-ENS 算法比 NSGA-II 算法节省了大量空间.

2) 时间复杂度.

假定两种算法具有相同的初始种群规模 N , 在每次迭代过程中, ADEMO/D-ENS 算法耗费时间主要集中在步骤 2 种群更新, 所需时间 $O(m \cdot NR \cdot N)$, NSGA-II 算法需要计算时间为 $O(m \cdot N^2)$, 两者所用时间比为 $O(m \cdot NR \cdot N) / O(m \cdot N^2) = O(NR) / O(N)$, 而 $NR \ll N$, 所以在相同配置条件下, ADEMO/D-ENS 算法的运算比 NSGA-II 算法运算速度快.

3.4 算法实现(Procedure)

ADEMO/D-ENS 算法程序实现步骤如下:

Step 1 初始化:

Step 1.1 设置各参数, 并产生 N 维权向量 $\lambda^i = (\lambda_1^i, \lambda_2^i, \dots, \lambda_m^i)$, $i = 1, 2, \dots, N$; 初始化邻域种群选择概率 $p_c = 1/n$, 确定邻域种群规模 NS , 计算与第 i 维权向量最近的 T 个权重索引集 $A(i) = \{i_1, \dots, i_T\}$;

Step 1.2 产生初始种群 $P^0 = \{x^1, \dots, x^N\}$, 并计算个体的适应度值 $FV^i = F(x^i)$; 初始化 $z^* = (z_1, \dots, z_m)$, $z_j = \min_{1 \leq i \leq N} f_j(x^i)$;

Step 1.3 设置策略选择概率 $p_k = 1/K$, 品质评估参数 $q = 0$;

Step 2 优化更新:

若不满足停止规则,

for $i = 1, \dots, N$, do

Step 2.1 确定差分操作范围 P ;

Step 2.2 对每一个父代 X^i , 基于 p_k 从策略池中

选择操作算子执行差分操作, 产生新解 y , 并用式(12)对 y 修正, 得到解 y_m ;

Step 2.3 由决策变量范围修正 y_m ;

Step 2.4 更新 z^* : 对每一个 $j = 1, \dots, m$, 从 P 中任选一个个体 k , 若 $g_Tch(y_m | \lambda^k, z^*) < g_Tch(x^k | \lambda^k, z^*)$, 则 y_m 替代旧解 x^k , 更新 z^* ;

Step 2.5 种群更新: 设置 $nr = 0$,

1) 若 $nr = NR$ 或 P 为空(即父代解没有被更新), 则输出 PS: $\{x^1, \dots, x^N\}$, PF: $\{F(x^1), \dots, F(x^N)\}$, 否则, 从 P 种群里随机选择 r_j ;

2) 如果 $g_Tch(y_m | \lambda^{r_j}, z^*) \leq g_Tch(x^{r_j} | \lambda^{r_j}, z^*)$, 则 $x^{r_j} = y_m$, $FV^{r_j} = F(y_m)$, $nr = nr + 1$, 从 P 中删除 r_j , 回到 1)。

Step 2.6 由式(7)–(9)更新策略选择概率 $p_k(g)$;

$g = g + 1$;

Step 2.7 对所有 NSs, 更新 $W_{c,g}$ 和 $W_{s,c,g}$, 如果 $\text{mod}(g, LP) == 0$, 由式(5)–(6)更新邻域种群选择概率 p_c ; 并对 $W_{c,g}$ 和 $W_{s,c,g}$ 重新置 0; 进入下一个 LP 周期.

Step 3 满足算法停止准则, 输出 PS: $\{x^1, \dots, x^N\}$, PF: $\{F(x^1), \dots, F(x^N)\}$.

4 仿真分析(Simulation analysis)

参考文献[1]设置部分仿真参数如下:

差分算法参数: 交叉概率 $CR = 1.0$, 缩放因子 $F = 0.5$, 插值概率 $P_{\text{inter}} = 0.75$, 多项式变异概率 $P_m = 1/n$, n 为决策变量维数, 多项式变异分布参数 $\tau = 20$.

PM 自适应算法参数: 自适应律 $\alpha = 0.3$, 每一个策略最小的选择概率 $p_{\min} = 0.05$.

变邻域分解算法参数^[4]: 邻域种群池 NSs = $\{30, 60, 90, 120\}$, 从邻域种群中选择父代解的概率 $\xi = 0.9$, 学习周期 $LP = 50$.

另外, 每个子代解替换父代解的最大个数 nr 随种群规模不同而变化, 设置 $nr = 0.01 \cdot N$.

选取 5 个测试函数: ZDT1, ZDT2, ZDT3, ZDT6 和 SCH.

4.1 参数敏感性分析(Parameter sensitivity analysis)

4.1.1 种群规模对算法的影响(Population size impact)

设置迭代次数 $G_{\max} = 500$, 种群规模分别为 $N = 300, 600$, 表 1 选择测试函数 ZDT1, ZDT2 和 ZDT3 仿真分析种群规模对算法性能的影响, 每个测试函数分别仿真 15 次.

表 1 结果说明, 除 ZDT2 的 IGD 方差外, 种群 $N = 600$, IGD 的均值与方差优于 $N = 300$ 的均值和方差, 说明种群 $N = 600$ 优化函数的 IGD 值在均值附近的分布比 $N = 300$ 的均值分布相对比较均匀, 但方差的

数量级较小, 为 10^{-7} 或 10^{-8} , 两种种群获得的Pareto占优解在均匀性分布上体现的不是很明显, 而均值的差别相对较大, 数量级为 10^{-3} , 对Pareto前沿的影响较大, 因此, 在两目标优化计算时, 通常采用 $N = 600$ 种群规模.

表 1 种群对算法性能的影响

Table 1 Population impact on the performance of the algorithm

测试函数	种群	均值	方差
ZDT1	300	$3.896\text{E} - 3$	$3.399\text{E} - 7$
	600	$2.663\text{E} - 3$	$1.473\text{E} - 7$
ZDT2	300	$3.245\text{E} - 3$	$9.077\text{E} - 8$
	600	$2.102\text{E} - 3$	$1.543\text{E} - 7$
ZDT3	300	$3.246\text{E} - 3$	$2.325\text{E} - 6$
	600	$2.039\text{E} - 3$	$5.394\text{E} - 7$

4.1.2 迭代次数对算法的影响(Iterations impact)

设置种群为 $N = 600$, 迭代次数为 $G_{\max} = 300, 500$, 表2选择测试函数ZDT2, ZDT3和ZDT6仿真分析迭代次数对算法性能的影响, 每个测试函数分别仿真15次.

表 2 迭代次数对算法性能的影响

Table 2 Iterations impact on the performance of the algorithm

测试函数	迭代次数	均值	方差
ZDT2	300	$1.996\text{E} - 3$	$2.1448\text{E} - 8$
	500	$2.102\text{E} - 3$	$8.205\text{E} - 8$
ZDT3	300	$2.458\text{E} - 3$	$8.111\text{E} - 7$
	500	$2.039\text{E} - 3$	$5.394\text{E} - 7$
ZDT6	300	$4.798\text{E} - 3$	$1.277\text{E} - 6$
	500	$5.075\text{E} - 3$	$2.792\text{E} - 6$

表2的结果说明迭代次数增加, ZDT2和ZDT6的性能指标参数均值和方差反而增大, 而ZDT3的性能指标参数的均值和方差随着迭代次数的增加均有一定的改善. 这说明, 对具体多目标优化函数求解Pareto最

优解时, 在对运算结果影响不大的前提下, 为改善算法的运算速度, 迭代次数可适当调整.

从表1和表2的结果, 可以发现, 改变种群规模和迭代次数对算法性能指标参数均有影响. 对不同的测试函数, 可适当调整算法的种群和迭代次数, 对于两目标优化问题, 通常设置种群规模 $N = 600$, 迭代次数 $G_{\max} = 500$.

4.2 仿真分析(Simulation analysis)

为使优化后目标函数的Pareto前沿更好的逼近真实前沿, 目标函数被评估300000次, 所以设置种群规模 $N = 600$, 最大迭代次数 $G_{\max} = 500$, 其他参数不变.

选取5个测试函数: ZDT1, ZDT2, ZDT3, ZDT6和SCH, 将IGD^[20]和 $C^{[3, 21]}$ 作为其性能指标参数, 每个测试函数分别仿真15次. 与经典多目标差分进化算法(MODE)和NSGA-II算法进行仿真对比, 说明本文算法的有效性和可行性. 设置MODE算法和ADEMO/D-ENS具有相同的差分操作控制参数, 差分策略选择“DE/rand/1”, 且3种算法具有相同的迭代次数和种群规模. 表3表示ADEMO/D-ENS(A)算法和MODE(B)算法得到的 C 指标参数.

表 3 $C(A, B)$ 性能指标参数

Table 3 $C(A, B)$ performance index parameter

测试函数	$C(A, B)$	$C(B, A)$
ZDT1	0.9867	0.0133
ZDT2	0.985	0.0133
ZDT3	0.9867	0.0317
ZDT6	0.9833	0.0117
SCH	0.9833	0.0117

表4表示不同算法得到的测试函数IGD性能指标参数, 图1-5为ADEMO/D-ENS(A)算法和MODE(B)算法对测试函数优化后的Pareto前沿和真实Pareto前沿, 其中, Pareto front 1表示ADEMO/D-ENS算法获得的最优Pareto前沿, Pareto front 2表示MODE算法获得的最优Pareto前沿.

表 4 迭代次数对算法性能的影响

Table 4 Iterations impact on the performance of the algorithm

测试函数	ADEMO/D-ENS算法		NSGA-II算法		MODE算法	
	均值	方差	均值	方差	均值	方差
ZDT1	$2.663\text{E} - 3$	$1.473\text{E} - 7$	$7.532\text{E} - 2$	$6.451\text{E} - 4$	0.231	$1.588\text{E} - 2$
ZDT2	$2.102\text{E} - 3$	$1.543\text{E} - 7$	0.2354	$1.985\text{E} - 3$	0.148	$5.162\text{E} - 3$
ZDT3	$2.039\text{E} - 3$	$5.394\text{E} - 7$	0.7538	$1.643\text{E} - 3$	0.817	$1.282\text{E} - 3$
ZDT6	$5.075\text{E} - 3$	$2.792\text{E} - 6$	4.021	0.247	$4.989\text{E} - 2$	$4.1665\text{E} - 4$
SCH	$1.028\text{E} - 2$	$3.813\text{E} - 6$	$1.457\text{E} - 2$	$2.036\text{E} - 3$	0.576	0.3098

表3说明MODE算法获得的Pareto解98%以上被ADEMO/D-ENS获得的Pareto解支配, 而ADEMO/D-ENS获得的Pareto解被MODE算法获得的Pareto解支配的最大比例仅为3.17%; 表4表明测试函数的均值和方差优于经典的NSGA-II算法和MODE算法的优化结果, 同时, 通过图1—5的对比可以看出, 虽然ADEMO/D-ENS算法和MODE算法都能够收敛到真实Pareto前沿, 但ADEMO/D-ENS算法得到的Pareto前沿的多样性和均匀性均优于MODE算法, 尤其对测试函数ZDT1, ZDT2和ZDT3的求解, 优化后的Pareto前沿在真实Pareto前沿上分布比较均匀, 而ZDT6测试函数的优化结果分布均匀性相对较差, 在 $f_1 \in (0.8, 1)$ 的范围内相对集中, SCH测试函数的优化Pareto前沿在两端极点结果相对较差, 其他部分的结果相对较好。

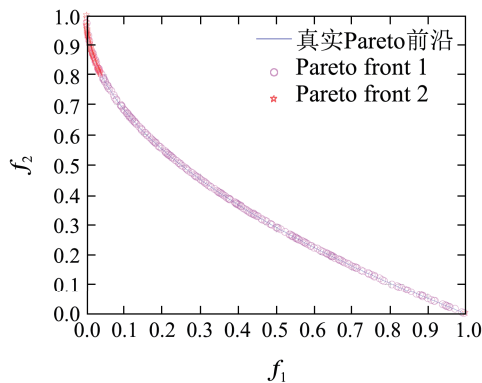


图 1 ZDT1真实前沿与ADEMO/D-ENS和MODE优化的Pareto前沿

Fig. 1 ZDT1 true front and optimized Pareto front by ADEMO/D-ENS and MODE

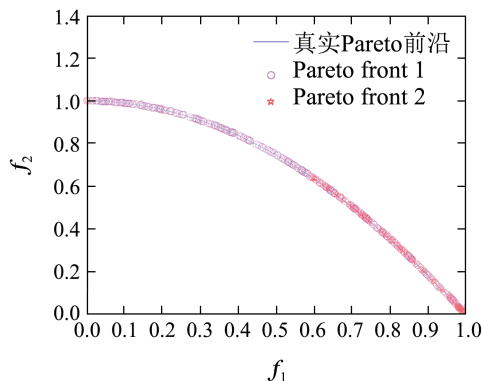


图 2 ZDT2真实前沿与ADEMO/D-ENS和MODE优化的Pareto前沿

Fig. 2 ZDT2 true front and optimized Pareto front by ADEMO/D-ENS and MODE

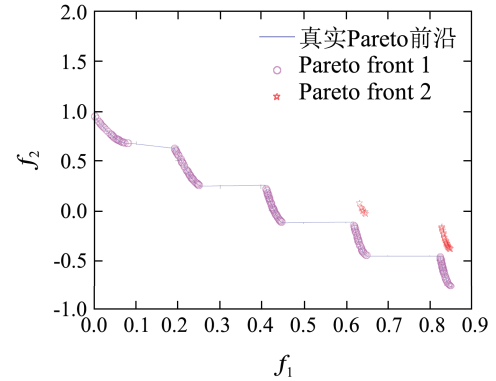


图 3 ZDT3真实前沿与ADEMO/D-ENS和MODE优化的Pareto前沿

Fig. 3 ZDT3 true front and optimized Pareto front by ADEMO/D-ENS and MODE

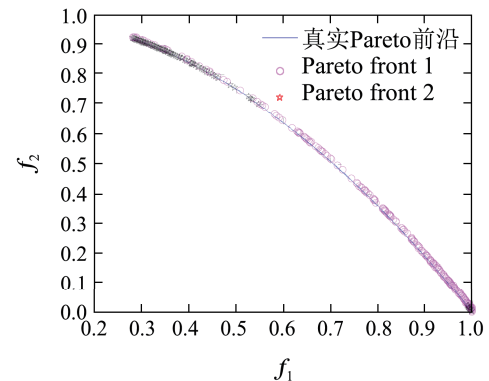


图 4 ZDT6 真实前沿与ADEMO/D-ENS和MODE优化的Pareto前沿

Fig. 4 ZDT6 true front and optimized Pareto front by ADEMO/D-ENS and MODE

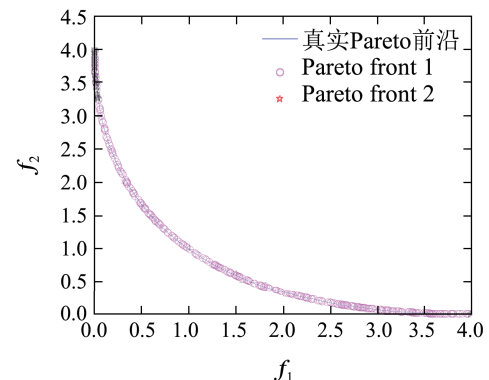


图 5 SCH真实前沿与ADEMO/D-ENS和MODE优化的Pareto前沿

Fig. 5 SCH true front and optimized Pareto front by ADEMO/D-ENS and MODE

图6—10为各测试函数性能指标IGD的box-plot图像, 除去箱盒外的异常值, 从图6—10可以看出, 除SCH函数外, ADEMO/D-ENS算法优化后的各测

试函数的IGD性能指标最大值均在 10^{-3} 数量级,比如ZDT1函数箱盒图中的最大值在 8.2621×10^{-3} ,最小值为 1.4304×10^{-7} ,而MODE算法得到的IGD性能指标最大值在0.3281,最小值在 1.4321×10^{-14} .这说明ADEMO/D-ENS算法得到的IGD指标具有更好的分布均匀性,这也印证了表4中的结果.

因此,说明和经典的NSGA-II算法和MODE算法相比,本文提出的ADEMO/D-ENS算法能够获得分布性和均匀性都更好的Pareto前沿,能够很好的解决多目标优化问题.

4.3 策略选择概率(Strategy selection probability)

本文设置4种差分策略组成差分策略池,图11-15为每个测试函数在进化过程中的差分策略选择概率变化曲线,经过多次仿真均证明迭代次数大约在20代时,所有策略选择概率趋于相等,为了更清楚的从图像上看到选择概率变化趋势,选择前100代的概率变化.

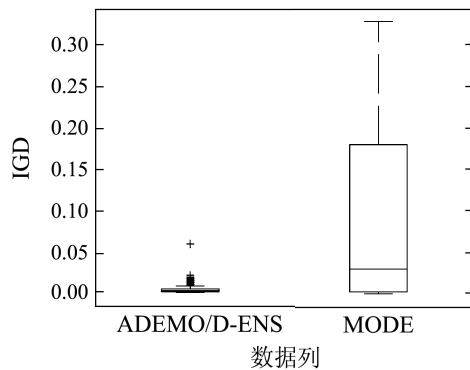


图6 ADEMO/D-ENS和MODE算法优化的ZDT1函数IGD指标箱盒图

Fig. 6 IGD boxplot of optimized ZDT1 function by ADEMO/D-ENS and MODE algorithm

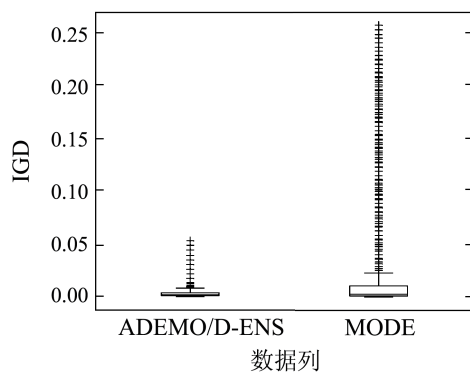


图7 ADEMO/D-ENS和MODE算法优化的ZDT2函数IGD指标箱盒图

Fig. 7 IGD boxplot of optimized ZDT2 function by ADEMO/D-ENS and MODE algorithm

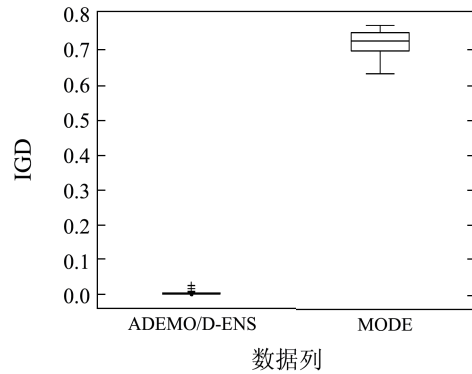


图8 ADEMO/D-ENS和MODE算法优化的ZDT3函数IGD指标箱盒图

Fig. 8 IGD boxplot of optimized ZDT3 function by ADEMO/D-ENS and MODE algorithm

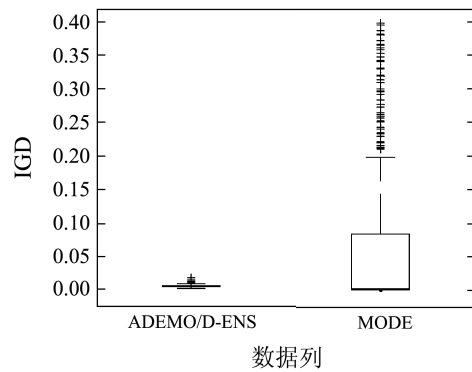


图9 ADEMO/D-ENS和MODE算法优化的ZDT6函数IGD指标箱盒图

Fig. 9 IGD boxplot of optimized ZDT6 function by ADEMO/D-ENS and MODE algorithm

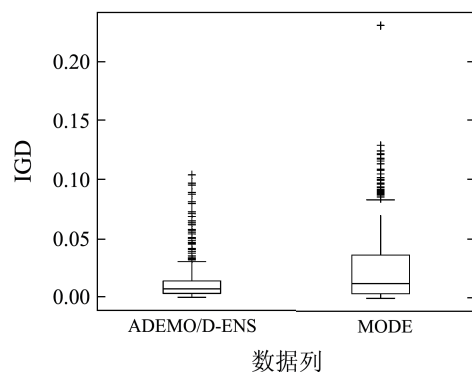


图10 ADEMO/D-ENS和MODE算法优化的SCH函数IGD指标箱盒图

Fig. 10 IGD boxplot of optimized SCH function by ADEMO/D-ENS and MODE algorithm

以测试函数ZDT1的策略选择概率变化曲线为例,分析说明在算法进化过程中,选择概率的变化情况.图11说明测试函数ZDT1在算法进化初期,策略1具有最高的选择概率,策略2被选中的概率最

小, 在第8代, 策略1的选择概率变化为最小, 策略3被选中的概率最大, 在第11代开始, 策略3和4的选择概率几乎相等, 策略2的选择概率变成最小, 直到第25代以后, 所有策略具有相等的选择概率. 从图12–15可以清晰的看到, 其他几个测试函数, 在进化算法初期, 不同的差分策略被选中的概率各异, 但在第25代以后, 所有测试函数的策略选择概率趋于相等.

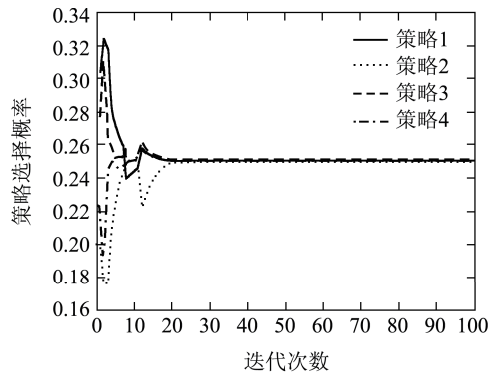


图 11 ZDT1差分策略选择概率变化曲线

Fig. 11 The curve of ZDT1 differential strategy selection probability

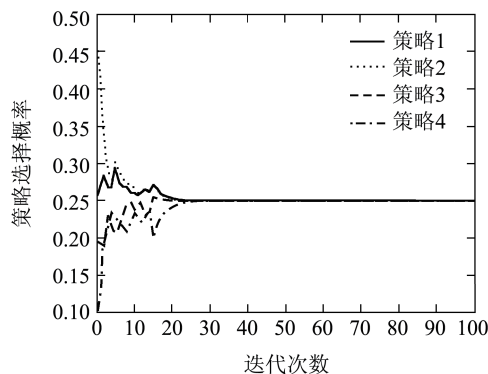


图 12 ZDT2差分策略选择概率变化曲线

Fig. 12 The curve of ZDT2 differential strategy selection probability

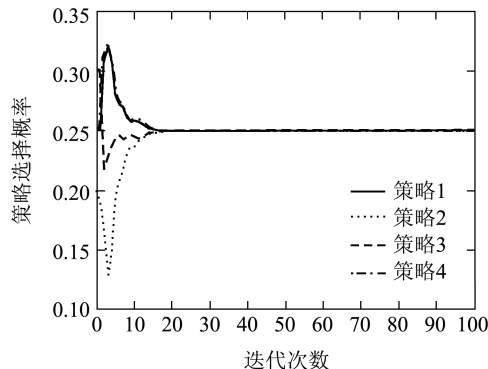


图 13 ZDT3差分策略选择概率变化曲线

Fig. 13 The curve of ZDT3 differential strategy selection probability

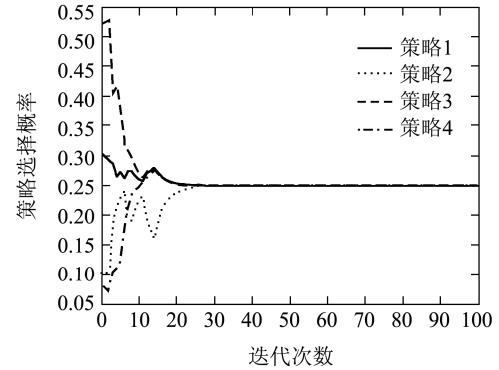


图 14 ZDT6差分策略选择概率变化曲线

Fig. 14 The curve of ZDT6 differential strategy selection probability

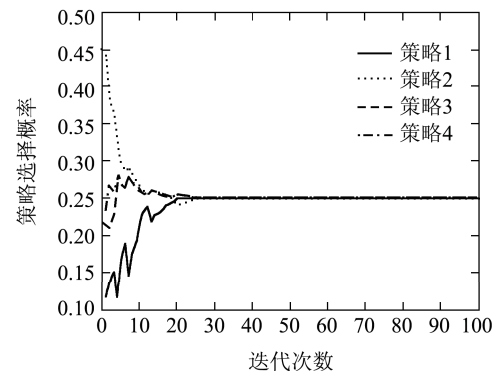


图 15 SCH差分策略选择概率变化曲线

Fig. 15 The curve of SCH differential strategy selection probability

4.4 邻域种群选择概率(Neighborhood size selection probability)

本文参考文献[4]设置了4个不同邻域规模组成邻域规模集合 $NSs=\{30, 60, 90, 120\}$, 图16–20说明在算法进化过程中邻域种群被选中的概率变化趋势.

图17说明, 在进化过程中, ZDT1测试函数选择邻域种群30的概率最高, 120的邻域种群具有最低的选择概率, 在算法进化到第461代时, 所有邻域种群具有相同的选择概率; 图18表明, ZDT2函数在算法操作初期, 60的邻域种群具有最高的选择概率, 120的邻域种群始终具有最低的选择概率, 而ZDT3和SCH测试函数, 在算法进化过程中, 30的邻域种群始终具有最低的选择概率, 而ZDT3在进化的前150代, 120的邻域种群具有最高的选择概率, 在150–350代期间, 60的邻域种群具有较高的选择概率; SCH函数在约110代后, 60的邻域种群选择概率基本在0.2478–0.251之间, 直到第461代, 所有邻域种群具有相等的选择概率. 图19说明, 在进化算法的前300, 除了操作初期, 30的邻域种群几乎具有最低的选择概率, 在310–461代, 120的邻域种群始终具有最高的选择概率, 461代后, 所有邻域种群具有相

等的选择概率. 从上述图像, 可以清楚的看到, 尽管进化过程中各邻域种群的选择概率各不相同, 但均在第461代后, 具有相等的选择概率.

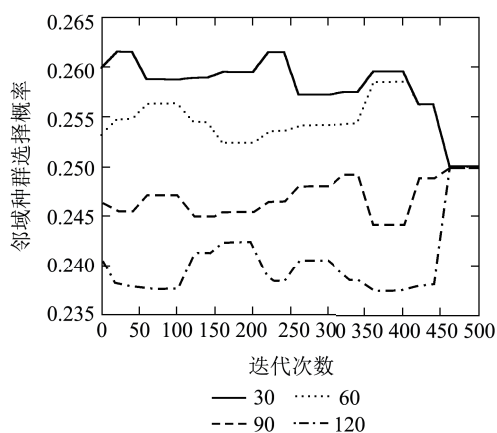


图 16 ZDT1邻域种群选择概率变化曲线

Fig. 16 The curve of ZDT1 neighborhood size selection probability

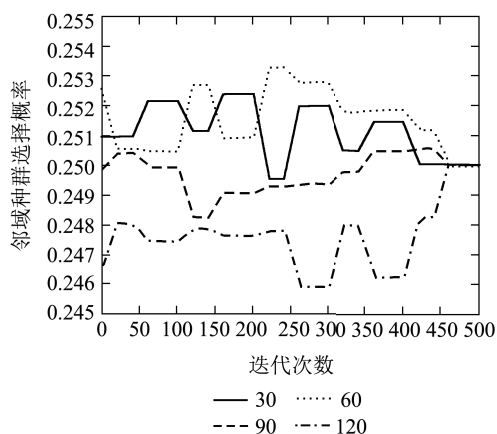


图 17 ZDT2邻域种群选择概率变化曲线

Fig. 17 The curve of ZDT2 neighborhood size selection probability

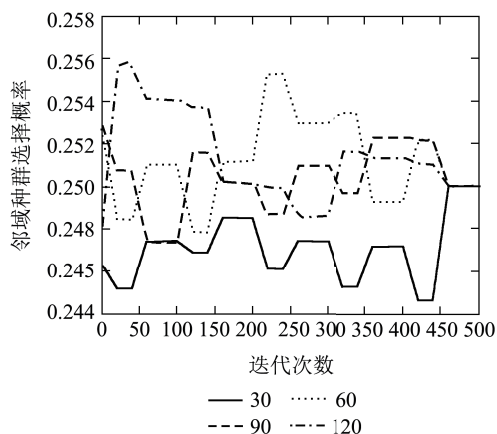


图 18 ZDT3邻域种群选择概率变化曲线

Fig. 18 The curve of ZDT3 neighborhood size selection probability

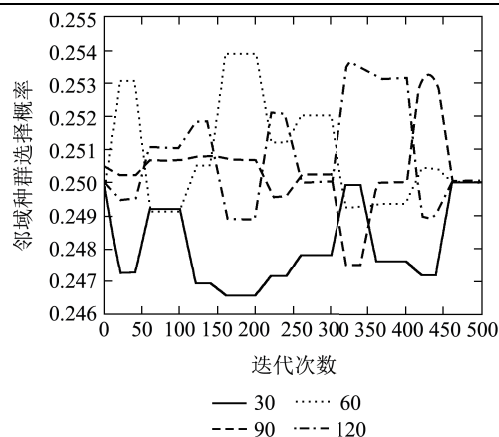


图 19 ZDT6邻域种群选择概率变化曲线

Fig. 19 The curve of ZDT6 neighborhood size selection probability

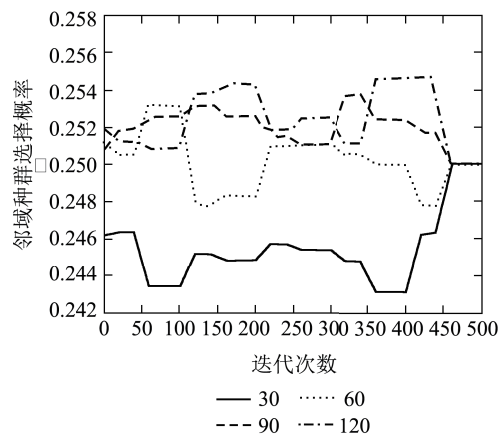


图 20 SCH邻域种群选择概率变化曲线

Fig. 20 The curve of SCH neighborhood size selection probability

5 结论(Conclusion)

本文提出了一种新颖的多目标优化算法——ADEMO/D-ENS, 该算法融合了基于邻域种群集的分解方法和自适应差分进化算法, 并采用概率自适应的方法选择邻域种群大小, 动态调整邻域种群的大小, 满足算法不同搜索阶段需要不同邻域种群的要求, 大的邻域种群帮助算法跳出局部最优, 小的邻域种群加快算法的收敛速度, 而待优化问题需要什么样的邻域种群集更合适是一个值得深入研究的问题; 同时采用PM方法, 从差分策略池中自适应选择差分变异策略, 满足算法不同搜索阶段需要不同差分策略的要求, 增加了可行解的多样性. 同时, 通过与经典NSGA-II算法的复杂度对比, 说明ADEMO/D-ENS算法的空间复杂度和时间复杂度均低于NSGA-II算法.

为验证本文提出算法的有效性和可行性, 采用了5个多目标优化标准测试函数, 并采用C和IGD

性能指标参数检验优化的Pareto前沿逼近真实Pareto前沿的程度, 并通过和经典的NSGA-II算法和MODE算法对比, 说明ADEMO/D-ENS具有更高的优化精度, 得到的Pareto优化前沿更逼近真实的Pareto前沿。

参考文献(References):

- [1] VENSKE S M, GONCALVES R A, DELGADO M R. ADEMO/D: multi-objective optimization by an adaptive differential evolution algorithm [J]. *Neurocomputing*, 2014, 127(3): 65 – 77.
- [2] DEB K. *Multi-objective Optimization Using Evolutionary Algorithms* [M]. Chichester: John Wiley & Sons, 2001.
- [3] ZHANG Q, LI H. MOEA/D: a multi-objective evolutionary algorithm based on decomposition [J]. *IEEE Transactions on Evolutionary Computation*, 2007, 11(6): 712 – 731.
- [4] ZHAO S Z, SUGANTHAN P N, ZHANG Q. Decomposition-based multi-objective evolutionary algorithm with an ensemble of neighborhood sizes [J]. *IEEE Transactions on Evolutionary Computation*, 2012, 16(3): 442 – 446.
- [5] CHIOU J P, CHANG C F, SU C T. Variable scaling hybrid differential evolution for solving network reconfiguration of distribution systems [J]. *IEEE Transactions on Power Systems*, 2005, 20(2): 668 – 674.
- [6] ADEYEMO J, OTIENO F. Differential evolution algorithm for solving multi-objective crop planning model [J]. *Agricultural Water Management*, 2010, 97(6): 848 – 856.
- [7] BABU B V, CHAKOLE P G, MUBEEN S J H. Multi-objective differential evolution (MODE) for optimization of adiabatic styrene reactor [J]. *Chemical Engineering Science*, 2005, 60(17): 4822 – 4837.
- [8] BABU B V, ANGIRA R. Modified differential evolution (MDE) for optimization of non-linear chemical processes [J]. *Computers & Chemical Engineering*, 2006, 30(6): 989 – 1002.
- [9] PRICE K, STORN R, LAMPINEN J. *Differential Evolution-A Practical Approach to Global Optimization* [M]. Berlin: Springer, 2005.
- [10] FAN H Y, LAMPINEN J. A trigonometric mutation operation to differential evolution [J]. *Journal of Global Optimization*, 2003, 27(1): 105 – 129.
- [11] CHAKRABORTY U K, DAS S, KONAR A. Differential evolution with local neighborhood [C] // *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*. Vancouver, BC: IEEE, 2006: 2042 – 2049.
- [12] DAS S, KONAR A, CHAKRABORTY U K. Improving particle swarm optimization with differentially perturbed velocity [C] // *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*. New York, USA: ACM, 2005: 177 – 184.
- [13] QIN A K, HUAN V L, SUGANTHAN P N. Differential evolution algorithm with strategy adaptation for global numerical optimization [J]. *IEEE Transactions on Evolutionary Computation*, 2009, 13(2): 398 – 417.
- [14] QIN A K, SUGANTHAN P N. Self-adaptive differential evolution algorithm for numerical optimization [C] // *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*. Edinburgh, Scotland: IEEE, 2005: 1785 – 1791.
- [15] GONG W, FIALHO A, CAI Z, et al. Adaptive strategy selection in differential evolution for numerical optimization: an empirical study [J]. *Information Sciences*, 2011, 181(24): 5364 – 5386.
- [16] THERIENS D. An adaptive pursuit strategy for allocating operator probabilities [C] // *Proceedings of the 2005 Conference on Genetic and Evolutionary Computation*. Washington, DC: ACM, 2005: 1539 – 1546.
- [17] SINDHYA K, RUUSKA S, HAANP T, et al. A new hybrid mutation operator for multi-objective optimization with differential evolution [J]. *Soft Computing*, 2011, 15(10): 2041-2055.
- [18] GOLDBERG D E. Probability matching, the magnitude of reinforcement, and classifier system bidding [J]. *Machine Learning*, 1990, 5(4): 407-425.
- [19] COELLO C A, LAMONT G B, VELDHIJZEN V D A. *Evolutionary Algorithms for Solving Multi-Objective Problems* [M]. New York: Springer, 2007.
- [20] ZHANG Q, ZHOU A, ZHAO S, et al. *Multi-objective optimization test instances for the CEC 2009 special session and competition* [R]. UK: University of Essex, 2008.
- [21] 雷德明, 严新平. 多目标智能优化算法及其应用 [M]. 北京: 科学出版社, 2009.
(LEI Deming, YAN Xinping. *Multi-objective Intelligent Optimization Algorithm and Its Application* [M]. Beijing: Science Press, 2009.)

作者简介:

刘志君 (1982-), 女, 博士研究生, 研究方向为控制系统性能指标分配与评估, E-mail: liuzhijun2001@163.com;

高亚奎 (1959-), 男, 兼职教授, 博士生导师, 研究方向为飞行器控制与系统仿真技术, E-mail: gaoyakui@126.com;

章卫国 (1956-), 男, 教授, 博士生导师, 主要研究方向为先进与智能飞行控制及优化技术、现代控制方法与智能控制方法及应用等, E-mail: zhangwg@nwpu.edu.cn;

王晓光 (1985-), 男, 博士研究生, 主要研究方向为飞机作战效能评估, E-mail: wangxiaoguang@nwpu.edu.cn;

袁燎原 (1988-), 男, 博士研究生, 主要研究方向为飞行控制系统传感器的容错方法设计, E-mail: yuanliaoyuan@163.com.